

Interrogazioni di tipo insiemistico

SQL fornisce anche gli operatori di tipo insiemistico

union, **intersect**, **except (minus)**

con la seguente sintassi:

SelectSQL

{ < **union** | **intersect** | **except** > [**all**] *SelectSQL* }

- **intersect** e **except** potrebbero anche essere derivati attraverso opportune query
- gli operatori insiemistici eseguono per default una eliminazione dei duplicati; **all** specifica di non farla
- gli schemi su cui si opera non devono essere identici come nell'algebra relazionale ma avere uguale numero di attributi e domini compatibili. La corrispondenza è per posizione.

Interrogazioni di tipo insiemistico

Estrarre nomi e cognomi degli impiegati

```
select Nome  
from Impiegato  
union  
select Cognome  
from Impiegato
```

Estrarre i cognomi degli impiegati che sono anche nomi

```
select Nome  
from Impiegato  
intersect  
select Cognome  
from Impiegato
```

Interrogazioni nidificate

E' possibile anche realizzare clausole **where** in cui il confronto non avviene fra predicati semplici o fra valori, ma fra valori e insiemi di valori, *appartenenti allo stesso dominio*, che sono risultato di un'altra interrogazione.

SQL offre 2 possibilità di estendere gli operatori di confronto al caso del confronto valore/insieme (risultato query):

all specifica che il risultato del confronto è vero se è vero per tutte le righe del risultato dell'interrogazione.

any specifica che il risultato del confronto è vero se è vero per una qualunque riga del risultato dell'interrogazione.

Interrogazioni nidificate

```
select I.Nome, Cognome  
from Impiegato I  
where Dipart = any (select D.Nome  
                     from Dipartimento D  
                     where Città='Firenze')
```

Il risultato comprende tutte le righe di IMPIEGATO per cui il valore Dipart è uguale ad almeno uno dei valori di Nome in DIPARTIMENTO, limitatamente alle tuple di DIPARTIMENTO per cui Città='Firenze',

cioè

l'elenco degli impiegati che lavorano in dipartimenti con sede a Firenze.

Interrogazioni nidificate

Lo stesso risultato si poteva ottenere con un join, ma così, specialmente per interrogazioni complesse, è più leggibile.

```
select I.Nome, Cognome
from   Impiegato I
where  Dipart = any (select D.Nome
                      from   Dipartimento D
                      where  Città = 'Firenze')
```

```
select I.Nome, Cognome
from      Impiegato I, Dipartimento D
where     Dipart = D.Nome
and       Città = 'Firenze'
```

Nome e reddito del padre di Franco

```
select Nome, Reddito
from   Persone, Paternita
where  Nome = Padre and
       Figlio = 'Franco'
```

```
select Nome, Reddito
from   Persone
where  Nome =(select Padre
               from Paternita
               where Figlio = 'Franco')
```

Interrogazioni nidificate

L'uso delle interrogazioni nidificate può anche eliminare la necessità degli alias.

Es. `select I1.Nome
 from Impiegato I1, Impiegato I2
 where I1.Nome = I2.Nome
 and I1.Dipart = 'Produz'
 and I2.Dipart <> 'Produz'`

equivale a

```
select Nome  
from Impiegato  
where Dipart = 'Produz'  
and Nome = any (select I.Nome  
                          from Impiegato I  
                  where I.Dipart<>'Produz')
```

Interrogazioni nidificate

Non tutte le interrogazioni nidificate corrispondono però ad un join

```
select Nome
from   Dipartimento
where  Nome <> all (select Dipart
                    from Impiegato
                    where Cognome='Rossi')
```

La condizione è verificata per le righe che NON contengono un certo valore, quindi non è esprimibile mediante un join, che richiede una corrispondenza fra valori. Però è equivalente a :

$$\Pi_{\text{Nome}}(\text{DIPARTIMENTO}) - \Pi_{\text{Dipart}}(\sigma_{\text{Cognome}='Rossi'}(\text{IMPIEGATO}))$$

e quindi le due query potevano essere unite da **except**

NB: **=any** e **<>all** si possono anche scrivere **in** e **not in**

Operatori aggregati e nidificazione

- La persona (o le persone) con il reddito massimo

```
select *  
from persone  
where reddito = ( select max(reddito)  
                  from persone )
```

Operatori aggregati e nidificazione

```
select  Dipart, Stipendio
from    Impiegato
where   Stipendio =
        ( select  max(Stipendio)
          from    Impiegato )
```

equivale a

```
select Dipart, Stipendio
from Impiegato
where Stipendio >= all ( select Stipendio
                        from Impiegato
```

(n.b. il massimo di un insieme è il valore, appartenente a quell'insieme, che è maggiore o uguale a tutti gli altri)

Operatori aggregati e nidificazione

Attenzione: un operatore aggregato non può essere applicato ad un altro operatore aggregato.

Ad es., non posso utilizzare **max** se voglio calcolare il massimo fra i risultati di un operatore aggregato applicato ai sottoinsiemi definiti con la clausola **group by**.

In questo caso è **obbligatorio** usare la definizione di massimo tramite un confronto fra valori di due sottoinsiemi (spesso uguali)

```
select Dipart, sum(Stipendio) as MaxSumSt
from Impiegato
group by Dipart
having MaxSumSt >=all (select sum(Stipendio)
                        from Impiegato
                        group by Dipart )
```